# HDF5: Managing Large Data Without Losing Your Mind

## Using the HDF5 File Format with R and Python

Jin Hyun Ju

Weill Cornell Graduate School of Medical Sciences

May 5, 2016

Caution: This talk may or may not be useful.

# What I do

Expression Quantitative Trait Loci (eQTL) Analysis

- Genotype Data = $N \times 800,000$
- Phenotype Data = $N \times 20,000$
- SNP information = Chromosome, Position, ids
- Gene information = Chromosome, Start, End, id
- Covariates

# Workflow

1. Phenotype vs Genotype association testing
2. Saving outcomes as a p-value matrix (around 200GB)
3. Calculating False Discovery Rate etc...
   (needs access to the whole p-value matrix)
4. Identify significant associations
5. Investigate significant associations using SNP and Gene information

# Challenges

- How can I keep information for a specific dataset in one place? (and minimize the chance of mixing up datasets...)
- Dealing with result files that are too big to load into memory at once
- Pass data between R and Python

# Primary Solution

- How can I keep information for a specific dataset in one place?
  Saved all the necessary objects in .RData files
  Can't inspect elements without loading them all into memory
  Only works with R
- Dealing with result files that are too big to load into memory at once
  bigmemory package: File backed matrices
  Only works with R
- Pass data between R and Python
  Temporary text files
  Unnecessary data duplication
  Uneasy feeling of creating waste

# Solution

Hierarchical Data Format 5 (HDF5)

- Folder like structures inside a single file
  Can hold various kinds of data
- Can get an overview of the contents
  Inspect elements without loading everything into memory
- Customizable structure
  Can create a structure that is convenient for the user
- Compression and chunking features available
  Fine tune I/O and storage space
- Fully compatible with R and Python
  Environment friendly